

インラインネットワーク計測技術のビデオ会議アプリケーションへの応用

長谷川 剛[†] 尾池 健二^{††} 天野 勝博^{††} 村田 正幸^{†††}

[†] 大阪大学 サイバーメディアセンター 〒560-0043 大阪府豊中市待兼山町1-32
^{††} ブラザー工業株式会社 NID 開発部 〒467-0855 愛知県名古屋市瑞穂区桃園町3番8号
^{†††} 大阪大学 大学院情報科学研究科 〒565-0871 大阪府吹田市山田丘1-5
E-mail: †hasegawa@cmc.osaka-u.ac.jp, ††{kenji.oike,katsuhiro.amano}@brother.co.jp,
†††murata@ist.osaka-u.ac.jp

あらまし ネットワークの輻輳状態などに応じた動画品質の調整機構は、これまでも多数の提案および製品が存在するが、それらのほとんどは、パケット廃棄や遅延時間を計測し、計測値の変化を契機にビットレート等の調整を行うため、ネットワーク特性や輻輳状態の変化に対する反応が本質的に遅れる。そこで本稿では、データ転送中のフローのパケットの送受信間隔を利用してパスの利用可能帯域を計測するインラインネットワーク計測手法を用いることにより、輻輳状態の初期変化を検知し、それに応じて、ビデオ会議アプリケーションが送受信する動画の品質を動的に調整する手法を提案する。計測手法の実験評価結果から、アプリケーション実装したインラインネットワーク計測手法が、ネットワークの利用可能帯域を高精度に計測可能であることを示す。

キーワード ネットワーク計測、利用可能帯域、ビデオ会議アプリケーション、動画品質

Application of inline network measurement technique to adaptive quality control in video conferencing system

Go HASEGAWA[†], Kenji OIKE^{††}, Katsuhiro AMANO^{††}, and Masayuki MURATA^{†††}

[†] Cybermedia Center, Osaka University 1-32, Machikaneyama-cho, Toyonaka, Osaka, 560-0043 Japan
^{††} NID Research and Development Dept. Brother Industries, Ltd. 3-8, Touen-cho, Mizuhoku, Nagoya, Aichi, 467-0855 Japan
^{†††} Graduate School of Information Science and Technology, Osaka University 1-5, Yamadaoka, Suita, Osaka, 565-0871 Japan
E-mail: †hasegawa@cmc.osaka-u.ac.jp, ††{kenji.oike,katsuhiro.amano}@brother.co.jp,
†††murata@ist.osaka-u.ac.jp

Abstract In this report, we propose the video conferencing system which change the audio and video quality according to the inline measurement results of the available bandwidth of the network path. We propose two types of application-level implementation methods for bandwidth measurement, which take different trade-off relationships between measurement accuracy and CPU load. From experimental results we confirm the effectiveness of the proposed measurement method.

Key words Network measurement, available bandwidth, video conferencing applications, video quality

1. はじめに

近年のインターネット環境の発展やユーザ数の爆発的な増加に伴い、様々なネットワークアプリケーションが登場している。特に、ネットワーク帯域の増加にともない、オンデマンドビデオ配信や実時間音声会話・ビデオ会議アプリケーションなどの、動画通信を伴うアプリケーションの利用が急速に進んでいる。ネットワーク帯域の増加にともない、これらのアプリケーショ

ンは音声・動画品質の向上や多地点対応などの高機能化が進められ、用いる帯域は増加する傾向にある。

従来このようなアプリケーションにおいては、ユーザが事前に動画や音声品質を指定するものがほとんどであった。その場合、ネットワークの利用可能帯域が小さくなった場合に音声や画質の途切れなどが発生し、アプリケーション品質が大きく劣化する。また逆に、ネットワーク帯域に余裕がある場合においても、それが認識できないと、ネットワーク帯域を効率的に

使うことができない。

そのため、ネットワーク帯域の使用効率を高め、アプリケーション品質を向上させるためには、ネットワーク特性や輻輳状況などを把握し、その様子に応じて動画品質などを調整し、アプリケーションが使うネットワーク帯域を制御することが重要となる。動画をを用いたネットワークアプリケーションの動的な品質制御については、これまでに多くの研究が行われている。それらの既存手法は、トランスポート層プロトコルとして User Datagram Protocol (UDP) を用い、アプリケーション動作と組み合わせることによって制御を行うもの [1-6]、および、トランスポート層として TCP を用い、輻輳制御手法を改変することによって制御を行うもの [7, 8] に大別される。また、ソフトウェアやアプライアンス機器として実現されているものとしては、ソニー社やポリコム社などの製品 [9-11] があり、ネットワーク状況を監視し、動画や音声のビットレートを変動させたり、誤り回復・制御手法である Automatic Repeat Request (ARQ) や Forward Error Correction (FEC) の制御パラメータ品質を動的に調整する機構を具備している。

上述の既存手法においては、送受信端末間のラウンドトリップ時間 (Round Trip Time: RTT) やパケット廃棄率などを計測することによってネットワーク特性や輻輳状態を監視し、その結果に応じてアプリケーションの動作を変更する。したがって、その制御はネットワークに輻輳が発生し、パケット廃棄率の上昇や遅延時間の増大を検出した後に行われるため、輻輳発生初期におけるパケット廃棄率や遅延時間の増大を避けることは本質的にできない。また逆に、パケット廃棄率の減少や遅延時間の低下に伴ってアプリケーションが用いる帯域を増加させる場合においても、その検出に時間がかかるため、帯域の利用効率が低下する。

これらの問題を解決するためには、パケット廃棄率や遅延時間の変動の直接的な原因である、ネットワーク帯域を直接的に把握することが有効であると考えられる。すなわち、アプリケーションが用いることができるネットワーク帯域に関する情報を計測によって獲得し、アプリケーション制御に利用することができれば、パケット廃棄や遅延時間の増大が発生する前にアプリケーショントラヒックの制御が可能となる。また、帯域に余裕が生まれた際にもその余剰帯域をすばやく使い切ることができるようになる。すなわち、帯域計測を利用することによって、ネットワーク特性や輻輳状態の変化への追随性を大きく向上させることができると考えられる。しかし、既存手法においては、そのような手法は用いられていない。これは、エンド端末間で用いることができるネットワーク帯域を高精度かつ低負荷で計測する手法がこれまで存在しなかったからであると考えられる。エンド端末間のネットワーク帯域を計測する手法は数多く提案されている [12-20] が、そのほとんどは計測のために長時間に渡って多くのパケットを高いレートでネットワークへ注入する必要があるため、計測トラヒック自体がネットワークに負荷を与え、アプリケーション品質の低下を招く。我々の研究グループにおいては、この問題を解消する手法として、インラインネットワーク計測手法 ImTCP を提案している [21]。ImTCP は、データ転送を行っている TCP コネクションのデータパケット及び ACK パケットを用いて、TCP コネクションが用いているエンド間パスの利用可能帯域を継続的に計測する (図 1)。ImTCP は計測用のパケットを必要としないため、ネットワークの他のトラヒックへの悪影響をほぼ完全に排除することができる。また、継続的な計測を行い、統計処理を行うことで、輻輳レベルの変化にともなう利用可能帯域の変動にすばやく追随することができる。しかしながら、TCP コネクション内での計測を行うため、RTP/RTCP 用いるビデオ会議アプリケーションには ImTCP を直接適用することができない。

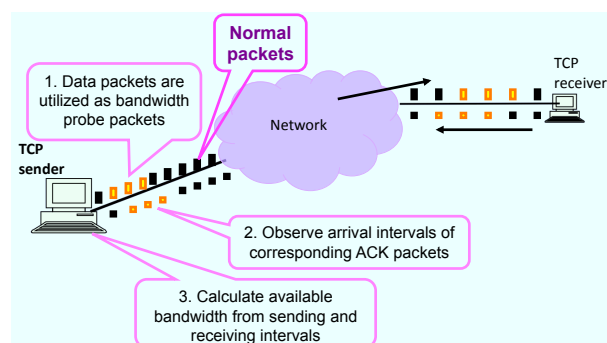


図 1 インラインネットワーク計測手法

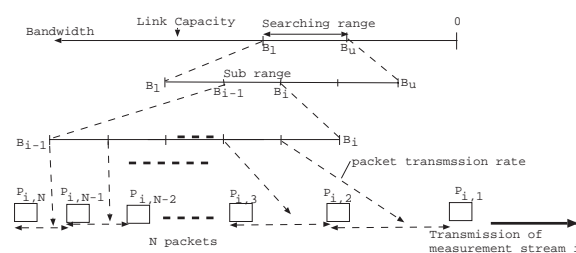


図 2 計測アルゴリズム

そこで本報告においては、インラインネットワーク計測手法を用いて利用可能帯域を計測し、計測結果に応じて動画のビットレートを変更することができるビデオ会議アプリケーションの設計を行う。また、計測手法の実装実験結果を示すことにより、その有効性を評価する。特に、TCP コネクション内で用いることを想定していた ImTCP を、アプリケーション層プログラムによって実現する際に発生する様々な問題に着目し、ImTCP が用いている計測アルゴリズムを実現することができるような計測用パケット群の構成方法などについて検討する。また、CPU へのオーバヘッドと計測精度のトレードオフに着目し、計測に必要なパケット送信間隔の実現方法として 2 種類の実装方法を提案し、その性能を比較評価する。性能評価に際しては、提案手法を実装し、実験ネットワーク環境における計測精度の評価を行い、提案方式の有効性を確認する。

以下、2. 章において、本稿において利用するインラインネットワーク計測技術について概説し、アプリケーションへの適用方法について議論する。3. 章では本稿において提案する、インラインネットワーク計測技術のビデオ会議アプリケーションへの適用方法について説明する。4. 章において提案手法の性能評価結果を示す。最後に 5. 章で本稿のまとめと今後の課題について述べる。

2. インラインネットワーク計測

本章では、[21] において我々の研究グループが提案している、インラインネットワーク計測手法の計測アルゴリズムについて説明し、そのアプリケーションへの適用方法について議論する。

2.1 計測アルゴリズム

提案方式は、送信側エンドホスト、受信側エンドホスト間の現在の利用可能帯域値を継続的に導出する。提案方式においては、まず送信側エンドホストが計測パケットを送出し、受信側エンドホストは受信した計測パケットをそのまま送信側エンドホストへ返送する。送信側端末は返送されたパケットの到着間隔から利用可能帯域の推測を行う。

提案方式において利用可能帯域を計測する際には、図 2 に示すように、現在の利用可能帯域値が含まれると考えられる帯域

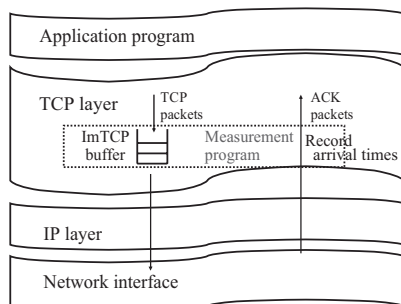


図3 ImTCPにおけるプロトコル階層図

の上限と下限を設定し、その間の探索区間の中から利用可能帯域を探す。探索区間を設定することで、不必要に高いレートで計測パケットを送出することを避けることができるため、外部トラフィックに与える影響を最小限に抑えることができ、計測パケット数を減少させることができる。提案する利用可能帯域計測アルゴリズムにおける、送信側エンドホストの動作概略を以下に示す。図2は、計測アルゴリズムの概略を示している。詳細については[21]を参照されたい。

(1) 初期探索区間の決定

まず、Cprobe[22]のアルゴリズムに基づいて初期探索区間を設定する。

(2) 探索区間の分割

探索区間を、大きさの等しい k 個の小区間に分割する。

(3) 計測ストリームの送出及びパケット間隔の比較

分割した k 個の小区間それぞれに対して、計測ストリームをネットワーク内に送出する。その際、1つの計測ストリームで様々な送信レートに対する計測を行うことで、計測パケット数を少なくし、短時間で計測結果を導出する。また、送出したストリームの受信結果から、PathLoad[12]のアルゴリズムを用いて、パケット間隔の増加傾向を調べる。

(4) 小区間の選択

全ての計測ストリームのパケット間隔の増加傾向から、現在の利用可能帯域値が含まれると考えられる小区間を選択する。

(5) 利用可能帯域を算出

ステップ(4)において探索区間内のある小区間に利用可能帯域が存在すると判断された場合は、線形回帰法を用いて利用可能帯域を導出する。探索区間内に利用可能帯域が存在しないと判断された場合には、利用可能帯域が大きく変化していると判断し、探索区間の両端の値を利用可能帯域とする。

(6) 探索区間の再計算を行い、(2)へ戻る

今回の計測結果と、前回までの計測結果を用いて、次回の計測で用いる探索空間を決定し、ステップ(2)へ戻る。

2.2 アプリケーションへの適用

[21]において我々は、上述の計測アルゴリズムをTCPに組み込むことで、データ転送中のTCPコネクションのデータパケット及びACKパケットを用いた計測手法を提案している。提案手法における計測アルゴリズムとプロトコルスタックの関係を図3に示す。図のように、TCP層の最下層部分にバッファ(ImTCPバッファ)を設け、TCPデータパケットをIP層へ引き渡す前に一旦蓄積し、計測アルゴリズムに従った送信間隔でパケットを送信する。また、ACKパケットを受信した際には、その到着時刻および到着間隔を記録し、計測アルゴリズムに従って利用可能帯域を算出する。このような仕組みにより、上位アプリケーションに対してネットワークパスの利用可能帯域に関する情報を提供することができる。

さらに我々の研究グループにおいては、ImTCPによる利用

可能帯域の計測結果を基に、従来のTCPでは提供できなかったトランスポート層サービスを実現するための輻輳制御手法を提案している。[23]においては、輻輳制御手法に数理生態学に基づくアルゴリズムを採用することにより、高速広帯域ネットワーク環境において高い性能を示すTCP Symbiosisを提案している。[24]においては、TCPコネクションが一定のスループットを確保できるような輻輳制御手法を提案している。また、[25]においては、競合トラフィックに影響を与えないデータ転送を行うバックグラウンド転送手法を提案している。

一方、OSのトランスポート層プロトコルを変更することが困難である状況を考慮して、アプリケーションプログラムによる計測アルゴリズムの実装および評価が行われている。[26]においては、UDPパケットを用いるアプリケーションプログラムによる実装実験結果が報告されている。また、[27]においては、ICMPパケットを用いた計測アルゴリズムの実装および性能評価結果が報告されている。しかし、具体的なアプリケーションを想定した検討や性能評価はこれまで行われていない。

トランスポート層の変更を行うことができない環境において、ビデオ会議システムのようなリアルタイム動画・音声アプリケーションに対してインラインネットワーク計測手法を適用する場合には、アルゴリズムをアプリケーション層において実現することが求められる。この場合、TCPへの組み込みとは異なる様々な点について考慮する必要がある。本稿においては、特に下記の点について着目した。

- TCPへの組み込みの際には、アプリケーションからバイトストリームが渡されるため、送信データが存在する限りは、最大サイズのデータパケットを送信することを前提とすることができた。一方、特にUDPを用いるアプリケーションを前提とし、アプリケーション層において実装を行う場合には、適用するアプリケーションが生成するデータグラムのサイズを考慮して計測用パケットを構成する必要がある。
- 計測パケットとして用いるデータパケットの送信間隔を調整する際に、CPU資源の利用量と計測精度のトレードオフを考慮し、どのような機構を用いるかを検討する必要がある。
- 一般的なリアルタイム動画・音声アプリケーションは、動画と音声で発生するパケットフローが異なる。そのため、それらのフローをどのようにバッファへ収容し、順次送信するかを決定する必要がある。
- TCPへの組み込みにおける図3のようなプロトコルスタックにおいて、特に一時的にパケットを蓄積するバッファをどのように実現するかを決定する必要がある。
- TCPへの組み込みにおいては、ネットワークを通過した後のパケット到着間隔として、送信側へ返送されるACKパケットの間隔を用いている。一方、アプリケーション層において実装を行う場合には、アプリケーション特性や送受信端双方の改変可能性に応じて、送信間隔および到着間隔をどちらの端末において集約して利用可能帯域の算出を行うかを決定する必要がある。

3.章において、これらの点を考慮した提案手法を説明する。

3. 提案手法

本章では、ビデオ会議システムにインラインネットワーク計測を適用する提案手法を説明する。

3.1 概要

図4に、提案システムの概要を示す。音声・映像コーデック、ネットワークモジュール、アプリケーション本体部分から構成されるビデオ会議アプリケーションに対して、インラインネットワーク計測を組み込むために必要となる、制御モジュール及び計測モジュールを追加している。計測モジュールは、アプリ

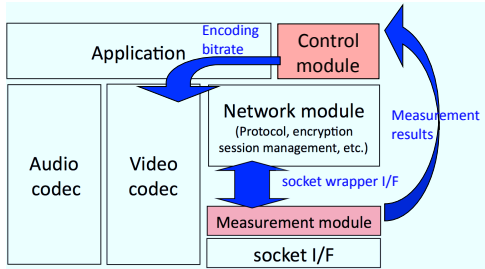


図 4 提案手法の概要

ケーションが行うプロトコル処理や暗号化処理による遅延の影響を避けるために、ソケットのラッパーとして動作する。そのため、ネットワークモジュールからは計測モジュールはソケットインタフェースの一部として認識される。計測モジュールは、計測結果を制御モジュールへ通知する。制御モジュールは受け取った計測結果を適宜処理し、コーデックに対して符号化レートを設定する。他のアプリケーションに対しても同様の構成を採用することによって、アプリケーションの変更を最小限に抑えつつ、インラインネットワーク計測を組み込むことができると考えられる。

以降ではメディアデータ（音声・動画データ）の送受信方法、計測用パケット群の構成方法について順に説明する。

3.2 メディアデータの送受信

送信側端末においては、音声および動画はそれぞれ個別のコーデックによって符号化され、パケット化されてネットワークモジュールへ渡される。ネットワークモジュールは宛先毎に用意しているセッションのソケットを通じて、計測モジュールへパケットを渡す。計測モジュールにおいては、音声および動画の packets はともに宛先毎に用意された FIFO バッファ(図 3 における ImTCP バッファに相当)にいったん蓄積される。計測モジュールにおいては、各 FIFO バッファに蓄積されたパケットの個数およびサイズに応じて、それぞれのパケット送信間隔を決定し、パケットをネットワークへ送出する。その際、パケット群の先頭部分には、送信間隔やシーケンス番号を書き込んだ制御パケットを付与し送信する。

受信側端末がメディアデータを受信した際には、計測モジュールは制御パケットの情報およびパケット群の到着時刻を記録し、制御パケット以外のパケットをネットワークモジュールへ渡す。ネットワークモジュールは受け取ったパケットを音声・動画ごとにコーデックへ受け渡すことで、受信データが再生される。計測モジュールは記録したパケット群の到着間隔および制御パケットに書き込まれている送信間隔を基に利用可能帯域を算出し、計測結果を制御モジュールへ受け渡す。計測結果は逆向きのセッションを介して送信側端末へ返送される。利用可能性帯域の計測結果を受信した送信側端末は、適宜統計処理を行うとともに、自身が起動しているコーデック数や通信相手数などを考慮し、各コーデックの符号化レートを決定し通知する。

3.3 計測用パケット群の構成方法

[21] において提案している計測アルゴリズムは、常に最大サイズのパケットが利用できることを前提とし、パケット毎に送信間隔を設定することで帯域計測を行っている。しかし、図 4 に示すビデオ会議システムにおいては、音声および動画の packets が独立して発生し、サイズも可変である。そのため、提案手法においては、複数のパケットをパケット群としてまとめて、[21] のアルゴリズムにおける 1 つのパケットとしてまとめて送信する。計測に用いるパケット群の合計サイズをそれぞれ S_1, S_2, \dots 、計測帯域をそれぞれ B_1, B_2, \dots 、および計測のために必要なパケット群間の時間間隔を T_1, T_2, \dots とする。 i 番目の

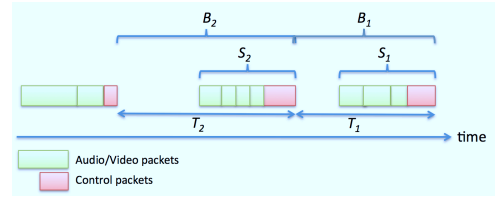


図 5 busy/wait を用いた計測パケット群の送出方法

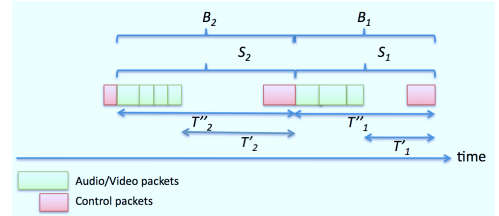


図 6 Sleep を用いた計測パケット群の送出方法

計測を行う際には、FIFO バッファに蓄積されている音声および動画の packets を複数個取り出し、 S_i に最も近くなれば、それに制御パケットを加えた合計のサイズが S_i になるように、制御パケットのサイズをダミーのペイロードを設定することによって調整する。このようにして生成した計測パケット群を、アルゴリズムに基づいた計測帯域を実現する時間間隔 T_i でネットワークへ送出する。

計測用のパケット群ストリームを実現するためには、構成した複数のパケット群の送信間隔を計測帯域値に応じて設定する必要がある。アプリケーションプログラムによって送信間隔を設定する方法には大きくわけて、(1) busy/wait を用いた手法、および (2) sleep を用いた手法、が考えられる。以下では、それぞれの場合におけるパケット群の構成方法について述べる。

3.3.1 busy/wait を用いた手法

busy/wait を用いた手法においては、計測モジュール内で busy/wait 命令を利用し、サイクル毎にクロック情報を確認することによって、指定した時間間隔が経過したか否かを判断する。この手法は、sleep を用いた手法に比べると時間間隔をより高い精度で実現できるため、利用可能帯域の計測精度が向上することが考えられる。そのため、図 5 において、 $S_1 = S_2 = \dots$ とし、 T_1, T_2, \dots を次の式のように設定する。

$$T_i = S_i / B_i \quad (1)$$

一方、指定した時間が経過するまで busy/wait を繰り返すことになるため、CPU 負荷が増大するという問題が存在する。

3.3.2 sleep を用いた手法

sleep を用いた手法においては、パケットを送出した後、sleep 命令を利用して指定した時間だけプログラムをスリープさせる。この手法は、スリープ中には CPU を消費しないため低負荷で実現することが可能であるという利点を持つが、コマンド実行時に指定できる時間粒度が OS によって限定される (Windows の場合には 10 msec) ことや、OS 上で同時に動いている他のプロセスの実行状況などの影響を受けるため、指定した時間のスリープが行えるとは限らない。

そのため、提案手法においては、sleep 命令によるスリープが終了した際に、実際にスリープしていた時間を計測し、その時間にに応じて、次に送出するパケット群のサイズを設定する。具体的には、図 6 において、 i 番目のパケット群の制御パケットを送出した後に行うスリープ動作から復帰した際にスリープしていた時間 T'_i を計測し、送出するパケット群のサイズ S_i を、 T'_i にパケット群を送出する時間を加えた値 T''_{ij} が以下の関係

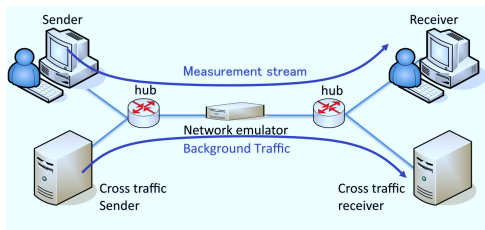


図7 実験環境

式を満たすように決定する。

$$S_i = B_i \cdot T_i'' \quad (2)$$

4. 性能評価

本章では、ビデオ会議システムへインラインネットワーク計測を組み込んだシステムを想定し、3.章において提案した計測手法をアプリケーションレベル実装を行い、実験網において性能評価を行った結果を示す。

4.1 評価環境

提案方式の実装は FreeBSD 7.2 及び Windows XP 上において、C 言語を用いて行った。本報告では Windows XP 上における実験結果を示す。実験ネットワーク環境を図7に示す。実験環境は提案方式を実装し、計測用ストリームを送受信する1対の送受信端末、背景トラフィックを送受信する送受信端末、2台のハブ、及びネットワークエミュレータから構成される。計測用ストリームを送信する端末は、Intel Core(TM)2 Duo 2.00GHz CPU 及び 980MB のメモリを搭載しており、Windows XP SP3 が稼働している。

ネットワークエミュレータは、ボトルネックリンクの帯域が 5Mbps になるように設定されている。背景トラフィックは、500 バイトの UDP パケットを用いて指定した帯域を消費するように送出される。提案手法においても、計測アルゴリズムの基本的な特性を検証するために、アプリケーションの packets ストリームを 800 バイトの UDP パケットを用いて構成している。また、sleep を用いた手法においては、利用可能帯域の計測結果の 80% の値をアプリケーションのデータ転送速度として用いる。

4.2 評価結果

図8に、busy/wait を用いる手法における計測結果を示す。また、背景トラフィックを5秒おきに变化させることによって、利用可能帯域が 1900 Kbps から 300 Kbps まで 200 Kbps 間隔で減少した後、再び 1900 Kbps まで増加させる、という変化を繰り返し発生させている。グラフには、計測結果 (measured)、移動平均 (average)、計測アルゴリズムにおける探索区間の上限 (upper) 及び下限 (lower)、及び真の利用可能帯域値 (true_bandwidth) を示している。

図より、真の利用可能帯域の値に関係なく、ほぼ正確に計測を行っており、かつ、帯域変化に対する追従性も問題ないことがわかる。しかし、利用可能帯域値が 500Kbps 以下の領域においては、追従性が若干低下している。これは、利用可能帯域が小さいために計測ストリームを送出する頻度が低下し、単位時間あたりの計測数が減少するためであると考えられる。

一方、計測実験中における送信側端末の CPU 利用率はおよそ 30% であった。提案しているビデオ会議システムは複数地点の同時参加も想定されており、複数の音声・動画コーデックが同時に起動されることもある。そのため、ビデオ会議システムに計測手法を統合した場合には、CPU 負荷が問題になるこ

とが考えられる。

図9に、sleep を用いる手法における計測結果を示す。また、背景トラフィックを1分おきに变化させることによって、利用可能帯域が 1900 Kbps から 300 Kbps まで 200 Kbps 間隔で減少した後、再び 1900 Kbps まで増加させる、という変化を繰り返し発生させている。グラフには、計測結果 (measured)、移動平均 (average)、及び真の利用可能帯域値 (true_bandwidth) を示している。

図より、sleep を用いた計測手法においても、高い計測精度が得られていることがわかる。アプリケーションのデータ転送速度を計測結果の 80% に設定しているにもかかわらず、利用可能帯域を正確に計測できているのは、インラインネットワーク計測手法がデータパケットの転送間隔を一時的に変更しているためである。しかし、真の利用可能帯域値が大きい場合に、計測結果の振動幅が大きいことがわかる。これは、sleep の時刻設定精度が粗いため、高い帯域を計測するために必要となる短い時間間隔を実現することが困難になるためである。

また、計測実験中における送信側端末の CPU 利用率はおよそ 0.21% であった。そのため、sleep を用いた手法は、複数地点が同時参加する状況においても問題なく計測が行えると考えられる。

5. おわりに

本報告においては、インラインネットワーク計測手法を用いて利用可能帯域を計測し、計測結果に応じて動画のビットレートを変更することができるビデオ会議アプリケーションの設計を行った。また、計測手法の実装実験結果を示し、提案した計測手法の効性を評価した。評価の結果、busy/wait を用いた手法は、計測精度は高いものの、高い CPU 負荷が問題になり得ること、また、sleep を用いた手法は、計測精度は若干低下するものの、CPU 負荷をほとんど増加させないことがわかった。

今後の課題としては、提案手法が持つ各種パラメータが与える影響の評価や、実ネットワーク環境における性能評価が挙げられる。

文 献

- [1] S. L. Bangolae, A. P. Jayasumana, and V. Chandrasekar, "TCP-friendly congestion control mechanism for an udp-based high speed radar application and characterization of fairness," in *Proceedings of the The 8th International Conference on Communication Systems*, pp. 164–168, 2002.
- [2] J. Lu and R. Ni, "Media streaming TCP-friendly congestion control using multiple time scale prediction," in *Proceedings of the Second International Conference on Innovative Computing, Informatio and Control*, p. 535, 2007.
- [3] L. Cai, X. Shen, J. Pan, and J. Mark, "Performance analysis of TCP-friendly AIMD algorithms for multimedia applications," *IEEE/ACM Transactions on Networking*, pp. 339–355, apr 2005.
- [4] F. Yang, Q. Zhang, W. Zhu, and Y.-Q. Zhang, "End-to-end TCP-friendly streaming protocol and bit allocation for scalable video over wireless Internet," *IEEE Journal on Selected Areas in Communication*, pp. 777–790, may 2004.
- [5] D. Deloddere, W. Verbiest, and H. Verhille, "Interactive video on demand," *IEEE Communication Magazine*, vol. 32, pp. 82–88, May 1994.
- [6] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-based congestion control for unicast applications," in *Proceedings of ACM SIGCOMM 2000*, pp. 43–56, Aug. 2000.
- [7] Y. Zhu, A. Velayutham, O. Oladeji, and R. Sivakumar, "Enhancing TCP for networks with guaranteed bandwidth

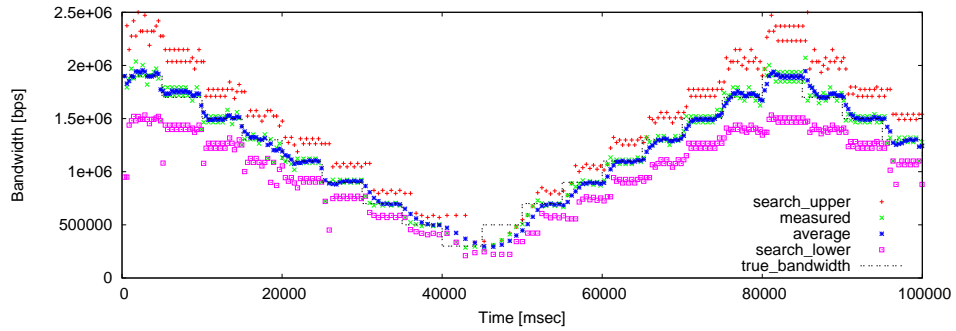


図 8 busy/wait を用いる手法の計測結果

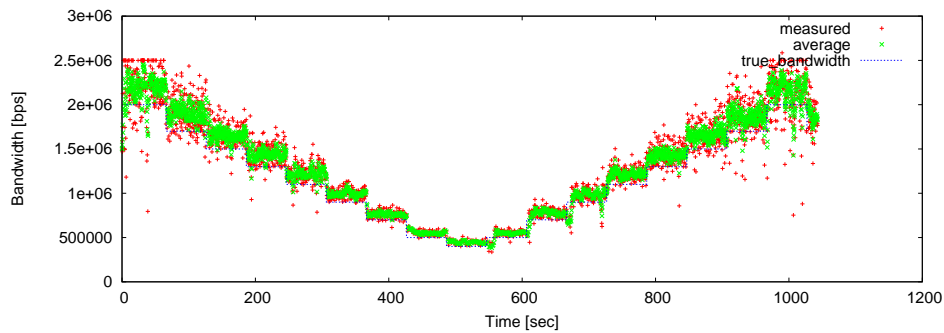


図 9 sleep を用いる手法の計測結果

- services,” *ACM Computer Networks: The International Journal of Computer and Telecommunications Networking*, vol. 51, pp. 2788–2804, July 2007.
- [8] T. Tsugawa, N. Fujita, T. Hama, H. Shimonishi, and T. Murase, “TCP-AFEC: An adaptive FEC code control for end-to-end bandwidth guarantee,” in *Proceedings of 16th International Packet Video Workshop (PV 2007)*, 2007.
- [9] Sony Corporation, QoS について. available at <http://www.sony.jp/pcs/technical/pdf/QoS.pdf>.
- [10] Sony Corporation, インテリジェント QoS. available at <http://www.sony.jp/pcs/technical/pdf/Intelligent-QoS2.V2.pdf>.
- [11] Polycom, Inc., Application-based quality of service for video conferencing. available at http://www.polycom.com/global/documents/whitepapers/application_based_quality_of_service.pdf.
- [12] M. Jain and C. Dovrolis, “End-to-end available bandwidth: Measurement methodology, dynamics, and relation with TCP throughput,” in *Proceedings of ACM SIGCOMM 2002*, Aug. 2002.
- [13] J. Strauss, D. Katabi, and F. Kaashoek, “A measurement study of available bandwidth estimation tools,” in *Proceedings of Internet Measurement Conference 2003*, Oct. 2003.
- [14] J. Navratil and R. Cottrell, “ABwE: A practical approach to available bandwidth estimation,” in *Proceedings of the 4th Passive and Active Measurement Workshop (PAM 2003)*, Apr. 2003.
- [15] V. Ribeiro, R. Riedi, R. Baraniuk, J. Navratil, and L. Cottrell, “PathChirp: Efficient available bandwidth estimation for network paths,” in *Proceedings of the 4th Passive and Active Measurement Workshop (PAM 2003)*, Apr. 2003.
- [16] R. Kapoor, L. Chen, L. Lao, M. Gerla, and M. Sanadidi, “CapProbe: A simple and accurate capacity estimation technique,” in *Proceedings of ACM SIGCOMM 2004*, Aug. 2004.
- [17] N. Hu and P. Steenkiste, “Evaluation and characterization of available bandwidth probing techniques,” *IEEE Journal on Selected Areas in Communications*, vol. 21, Aug. 2003.
- [18] J. Strauss, D. Katabi and F. Kaashoek, “A measurement study of available bandwidth estimation tools,” in *Proceedings of the Internet Measurement Conference*, 2003.
- [19] R. Anjali, C. Scoglio, L. Chen, I. Akyildiz, and G. Uhl, “ABEst: An available bandwidth estimator within an autonomous system,” in *Proceedings of IEEE GLOBECOM 2002*, Nov. 2002.
- [20] Sprobe available at <http://sprobe.cs.washington.edu>.
- [21] L. T. M. Cao, G. Hasegawa, and M. Murata, “ImTCP: TCP with an inline measurement mechanism for available bandwidth,” *Computer Communications Journal special issue of Monitoring and Measurements of IP Networks*, vol. 29, pp. 1614–1626, June 2006.
- [22] R. Carter and M. Crovella, “Measuring bottleneck link speed in packet-switched networks,” Tech. Rep. TR-96-006, Boston University Computer Science Department, Mar. 1996.
- [23] G. Hasegawa and M. Murata, “TCP symbiosis: congestion control mechanisms of TCP based on Lotka-Volterra competition model,” in *Proceedings of Inter-Perf 2006*, Oct. 2006.
- [24] G. Hasegawa, K. Yamanegi, and M. Murata, “TCP congestion control mechanisms for achieving predictable throughput using inline network measurement,” *IEICE Transactions on Communications*, vol. E91-B, pp. 3945–3955, Dec. 2008.
- [25] T. Tsugawa, G. Hasegawa, and M. Murata, “Background TCP data transfer with inline network measurement,” *IEICE Transactions on Communications*, vol. E89-B, pp. 2152–2160, Aug. 2006.
- [26] Insixiangmai Leuth, “サービスオーバレイネットワークのためのインラインネットワーク計測手法の実装および評価,” *卒業研究報告*, Feb. 2003.
- [27] 織田弘樹, 久松潤之, “ICMP を用いた高速かつ低負荷な利用可能帯域計測手法,” *電気学会通信研究会*, June 2010.